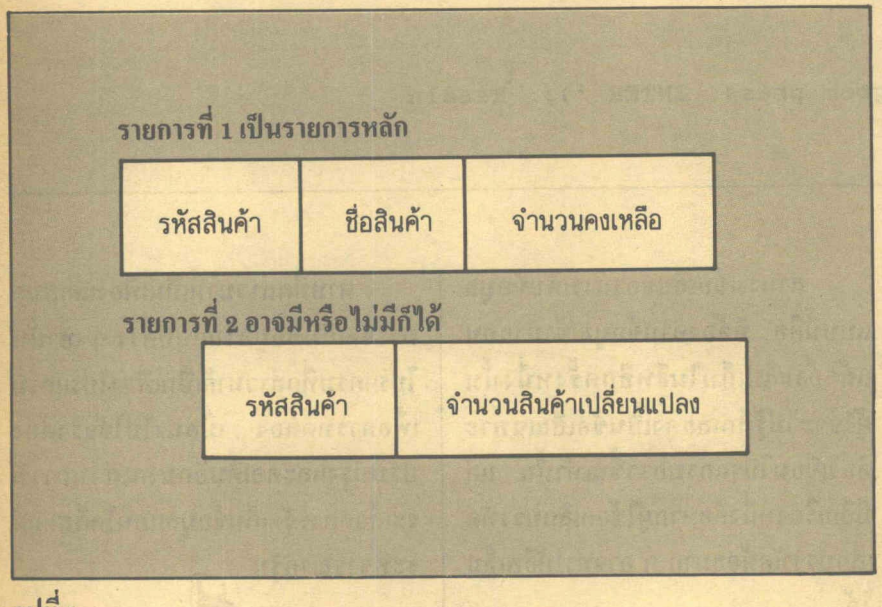
เทคนิคการออกแบบโปรแกรม
ตอน 6 (จบ)

อาตาปี & VIP

ในวารสารไมโครฯ ฉบับที่ 46 ได้กล่าวถึงผังโปรแกรมโครงสร้างแบบแจ็คสัน (Jackson Structured Programming) ในส่วนของรูปแบบและส่วนประกอบของข้อมูล สำหรับในฉบับนี้จะกล่าวถึงเทคนิคการวิเคราะห์ข้อมูลการออกแบบโปรแกรมตามหลักการของแจ็คสัน การสังเคราะห์โปรแกรม รวมถึงเรื่องโครงสร้างที่ขัดแย้งกัน (Structure Clashes)



รูปที่ 1

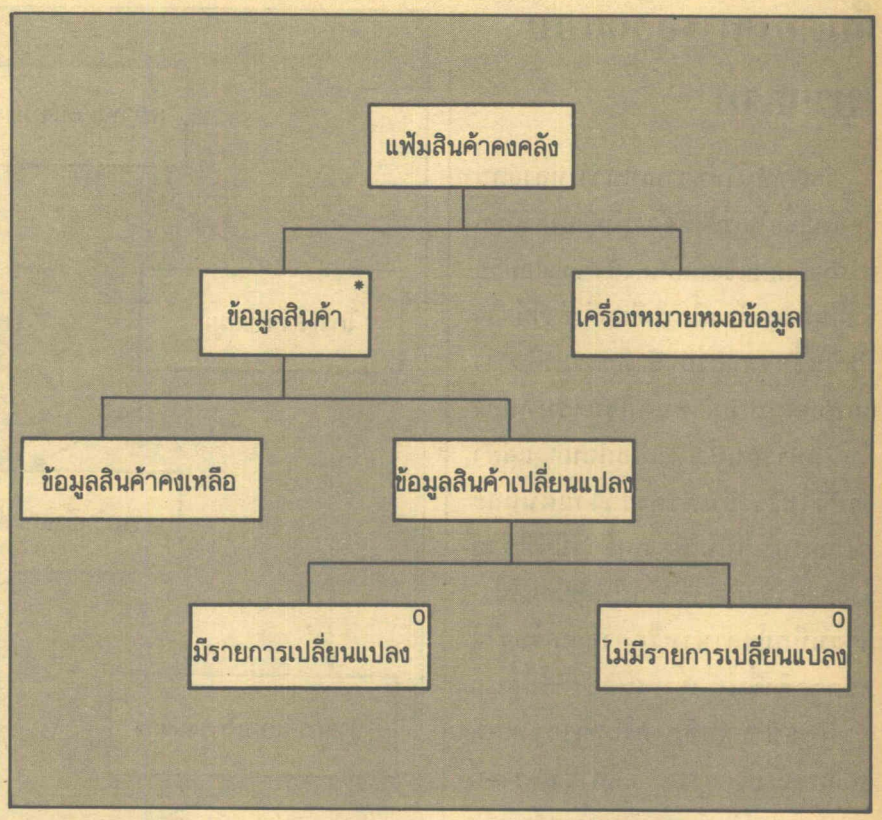
เทคนิคการวิเคราะห์ข้อมูล

การวิเคราะห์ข้อมูลทั้งส่วนข้อมูลนำเข้า ส่วนข้อมูลนำออกและส่วนของโปรแกรม มีหลักวิธีวิเคราะห์ได้ 2 วิธี คือ การพิจารณาจากลำดับชั้นกับความสำคัญของข้อมูล และการพิจารณาจากกายภาพกับองค์ประกอบของข้อมูล

วิธีที่ 1 การพิจารณาจากลำดับชั้นและความสำคัญข้อมูล

ตัวอย่างการวิเคราะห์โครงสร้างข้อมูลส่วนนำเข้าจากข้อมูลสินค้าคงคลัง ซึ่งมีรูปแบบข้อมูลส่วนนำเข้าดังรูปที่ 1

แจ็คสันได้มองภาพข้อมูลเป็นโครงสร้างได้ด้วยการจัดลำดับชั้นตามความสำคัญของข้อมูล และแยกย่อยออกตามสภาพเป็นแบบไปตามลำดับ เลือก หรือซ้ำ ตามตัวอย่างนี้เริ่มต้นจากแฟ้ม แยกย่อยเป็นตัวข้อมูลที่เกิดซ้ำ ๆ และเครื่องหมายหมดข้อมูล ต่อจากนั้นพิจารณาข้อมูลที่เกิดซ้ำว่าจะแตกออกเป็นข้อมูลอะไรต่อไป ในที่นี้แยกได้เป็นข้อมูล 2 กลุ่มเรียงความสำคัญตามลำดับคือ กลุ่มข้อมูลหลักหรือข้อมูลสินค้าคงเหลือ และกลุ่มข้อมูลเปลี่ยนแปลง สำหรับข้อมูลสินค้าเปลี่ยนแปลงมีสภาพที่อาจมีรายการหรือไม่มีก็ได้ ดังนั้นจึงแยกระดับข้อมูลลงไปอีกชั้นเป็นโครงสร้างเลือกคือ มีข้อมูล หรือไม่มีข้อมูล โครงสร้างเขียนได้ดังรูปที่ 2

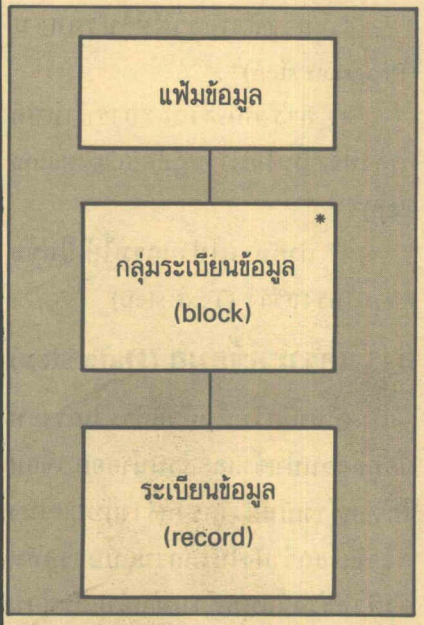


รูปที่ 2

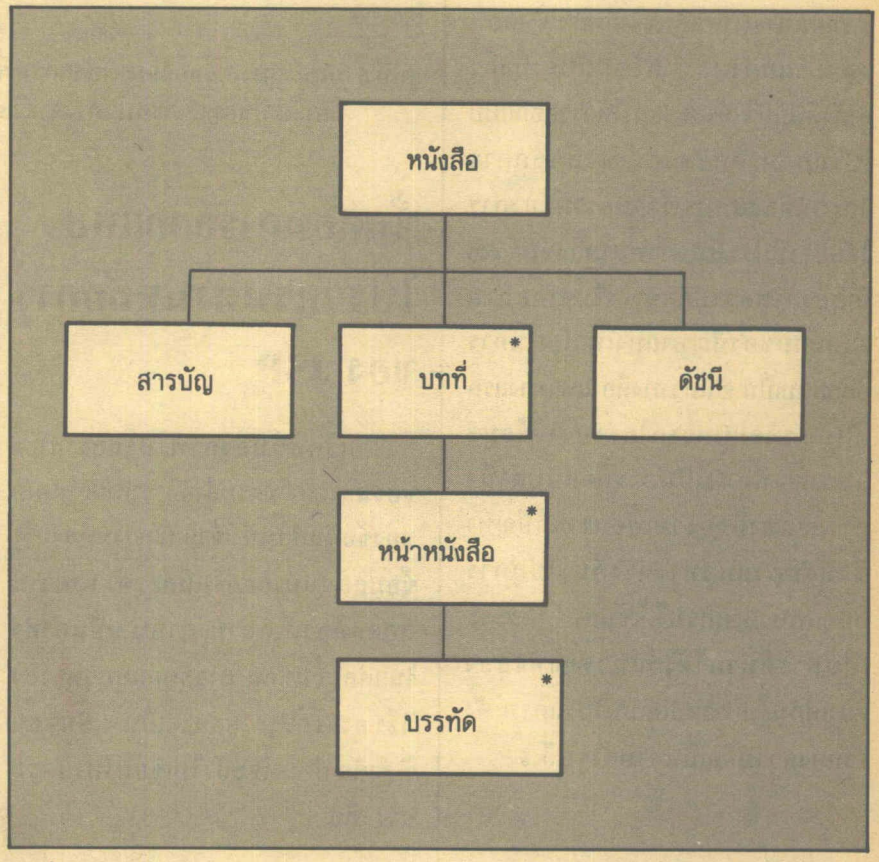
วิธีที่ 2 การพิจารณาโครงสร้างข้อมูลจากกายภาพและองค์ประกอบ

ตัวอย่างองค์ประกอบของแฟ้มข้อมูลที่บันทึกอยู่ในเทปแม่เหล็กเขียนเป็นโครงสร้างดังรูปที่ 3

ตัวอย่างกายภาพและองค์ประกอบของหนังสือทั่ว ๆ ไป ดังรูปที่ 4



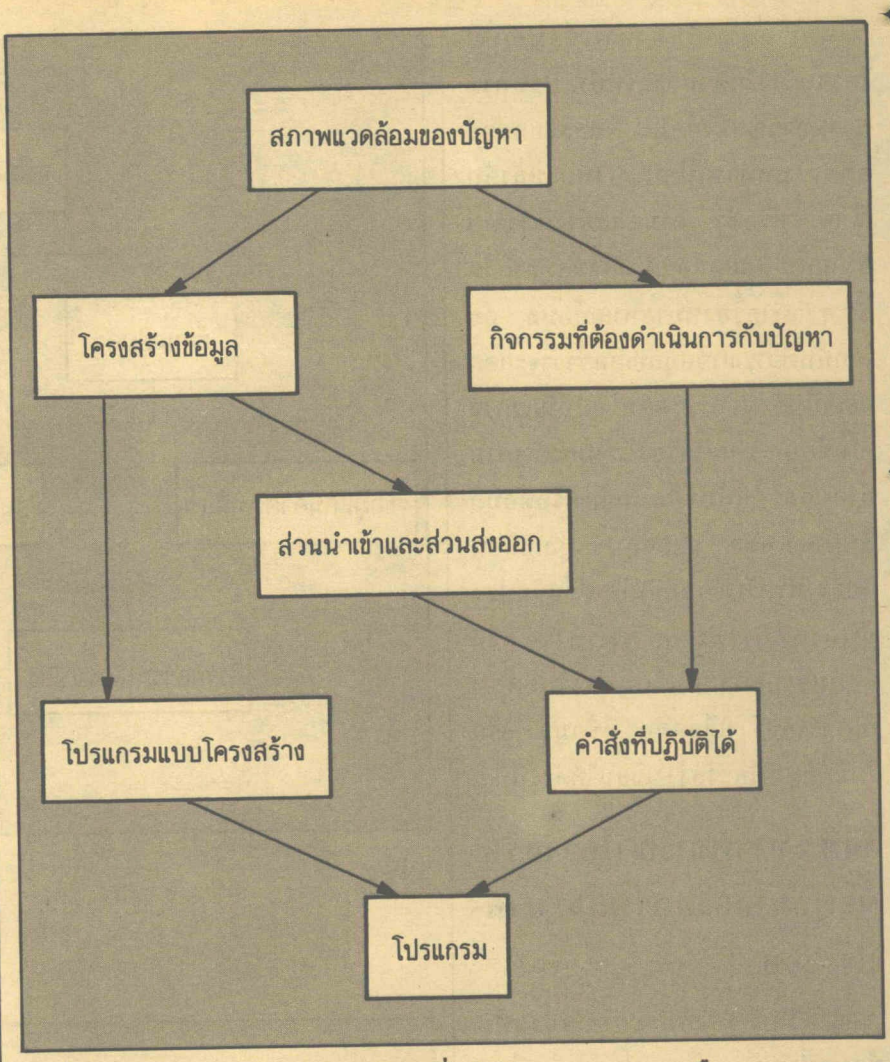
รูปที่ 3



รูปที่ 4

ขั้นตอนการออกแบบโปรแกรม

แจ็คสันให้ความเห็นว่า หลายคนมีความเชื่อในเรื่องที่ต้องทำผังคำสั่งก่อนการถอดแบบเขียนเป็นคำสั่ง เขายอมรับว่าเป็นสิ่งถูกต้องที่การเขียนคำสั่งไม่ควรทำเป็นขั้นตอนแรก สิ่งที่ควรทำคือวางแผนออกแบบกำหนดกิจกรรมที่ต้องดำเนินการกับปัญหาเสียก่อน แต่ผังคำสั่งไม่ใช่เป็นตัวการวางแผนและออกแบบ เป็นเพียงเครื่องมือที่ช่วยจำลองความคิดและช่วยให้ผู้เขียนโปรแกรมนึกถึงภาษาหรือคำสั่งเครื่องได้สะดวกยิ่งขึ้น เขาเสนอว่ามีวิธีอื่นที่ช่วยเพิ่มประสิทธิภาพในการวางแผนออกแบบโปรแกรม คือการวิเคราะห์จากโครงสร้างข้อมูลเป็นเบื้องต้นแล้วนำมาถอดออกเป็นผังโปรแกรมอย่างเป็นโครงสร้างแบบแขนง ประกอบกับการพิจารณาลำดับการนำเข้าสู่การประมวลผลการนำผลลัพธ์ในแต่ละขั้นตอนออกมาแสดงผล พร้อมกับคำนึงถึงคำสั่งที่ปฏิบัติได้จะช่วยทำให้การออกแบบโปรแกรมได้สะดวกอย่างมีหลักการและถูกต้องสมบูรณ์ตามความต้องการได้อย่างมีประสิทธิภาพมากยิ่งขึ้น เขาจำลองภาพความคิดนี้ เริ่มจากสภาพแวดล้อมของปัญหาแบ่งเทคนิคในการออกแบบเป็น 2 แนวทางคือ แนวทางแรกเป็นการคิดแบบจากโครงสร้างข้อมูล แนวทางที่สองเป็นการคิดแบบคำนึงการแยกหากิจกรรมเพื่อการแก้ปัญหา ซึ่งทั้งสองแนวทางต่างก็นำไปสู่การออกแบบโปรแกรมได้เช่นกัน และจะเป็นการดีหากได้ใช้แนวทางทั้งสองควบคู่กันในการออกแบบโปรแกรม ดังจำลองความคิดเป็นภาพดังรูปที่ 5



รูปที่ 5 แสดงแนวการออกแบบโปรแกรมที่แบ่งเป็นสองแนวทาง คือ แนวทางวิเคราะห์ข้อมูลและแนวทางการวิเคราะห์กิจกรรมเพื่อแก้ปัญหา

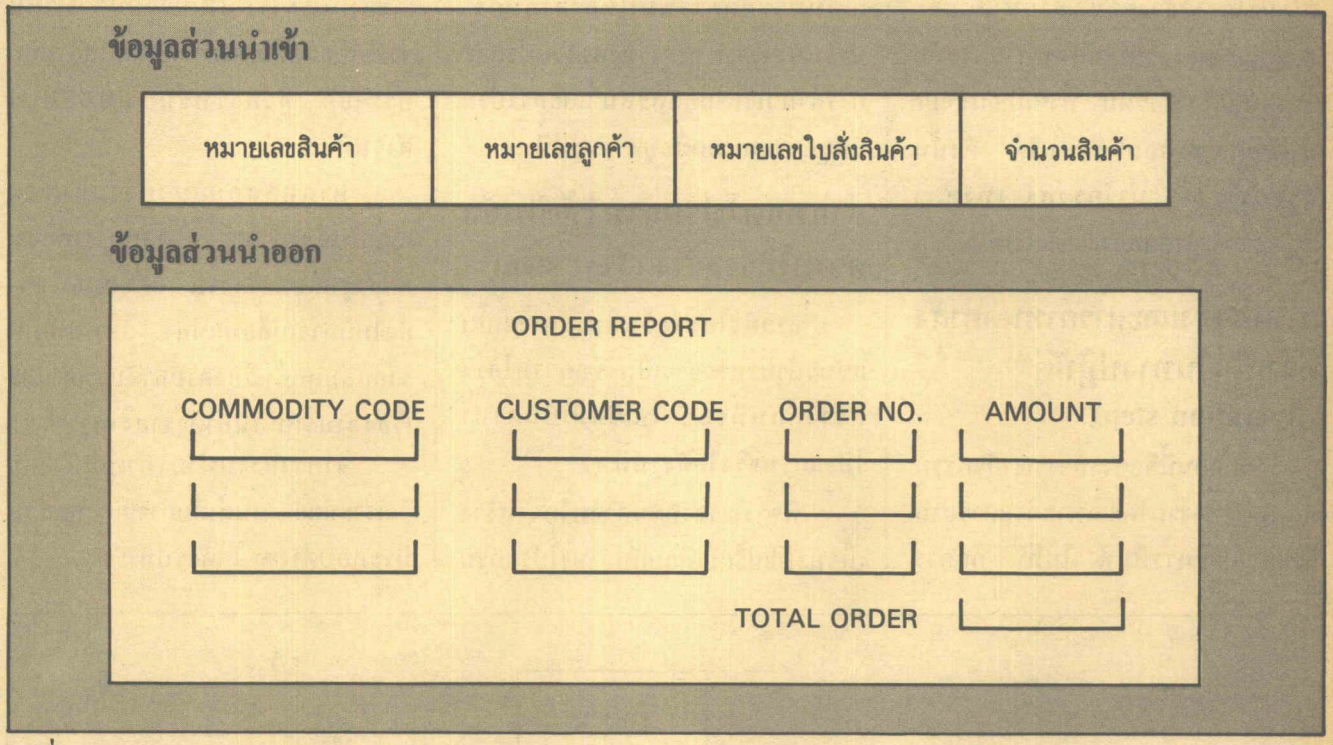
ขั้นตอนการออกแบบโปรแกรมตามหลักการของ JSP

แจ็คสันมองภาพโปรแกรมเป็นขบวนการทำไปตามลำดับ จากข้อกำหนดของข้อมูลส่วนนำเข้าและความต้องการข้อมูลส่วนนำออกไปที่ละจุด หาความสอดคล้องเพื่อนำมากำหนดเป็นคำสั่งในแต่ละขั้นตอน การออกแบบชุดคำสั่งวิธีของแจ็คสัน แยกได้เป็น 4 ขั้นตอนที่สำคัญก่อนที่จะนำไปเขียนโปรแกรมจริง คือ

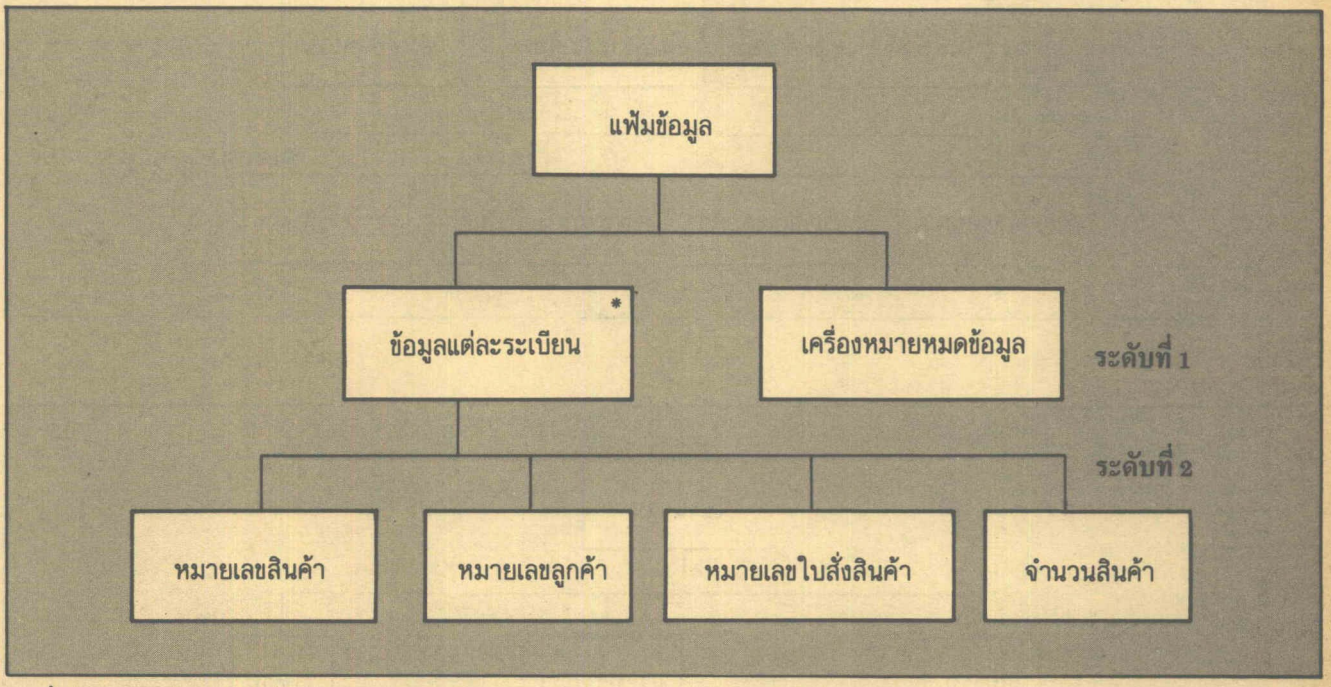
1. การวิเคราะห์ข้อมูล (Data step)
2. การสังเคราะห์โปรแกรม (Program step)
3. การตกแต่งและการกำหนดคำสั่งที่จำเป็นในทางปฏิบัติ (Operation step)
4. ถ่ายทอดโปรแกรมให้เป็นข้อความโครงสร้าง (Text step)

การวิเคราะห์ข้อมูล (Data step)

เป็นขั้นเริ่มต้นด้วยการวิเคราะห์ข้อมูลส่วนนำเข้าและส่วนนำออก เขียนโครงสร้างเป็นผังโครงสร้างแบบแขนง หรือที่เรียกว่าผังโปรแกรมแบบแจ็คสัน ดังตัวอย่างข้อมูลส่วนนำเข้าและส่วนนำออกดังรูปที่ 6



รูปที่ 6



รูปที่ 7

จากข้อมูลส่วนนำเข้าใช้วิธีวิเคราะห์องค์ประกอบ ทำเป็นผัง JSP แบ่งได้เป็น 2 ระดับได้ดังรูปที่ 7

วิเคราะห์โครงสร้างของข้อมูลส่วนนำออก ลองใช้วิธีลำดับความสำคัญของการเกิดและรูปแบบทำเป็นผังได้รูปที่ 8

การสังเคราะห์โปรแกรม (Program step)

นำโครงสร้างข้อมูลที่วิเคราะห์ได้นำมาเปรียบเทียบกัน เพื่อสังเคราะห์เป็นโครงสร้างของชุดคำสั่ง การเปรียบเทียบดูจากขั้นตอนการนำเข้า-ส่งออกรับกันหรือไม่ ระดับของชั้นข้อมูลมีระดับชั้นเท่ากันหรือไม่ จากตัวอย่างข้างต้น

เมื่อนำโครงสร้างข้อมูลส่วนนำเข้าและส่วนส่งออก มาเทียบกันพบว่ามีระดับชั้นเท่ากันคือ 2 ระดับ ทั้งขั้นตอนข้อมูลนำเข้าและส่งออกก็รับกันได้ ดังนั้นสรุปว่าใช้ได้ นำโครงสร้างทั้งสองสังเคราะห์เข้าด้วยกันได้ดังรูปที่ 9

การตกแต่งและการกำหนดคำสั่งที่จำเป็นในทางปฏิบัติ (Operation step)

ขั้นตอนนี้คือการพิจารณากิจกรรมที่เน้นการสั่งงานโดยเฉพาะด้านการอ่านคำนวณและการพิมพ์ ในที่นี้ คือการเลื่อนบรรทัดการพิมพ์หลังจากมีกิจกรรมพิมพ์ต่าง ๆ การตั้งค่าตัวแปรและการคำนวณหายอดสะสม และการอ่านข้อมูลซ้ำจนหมดข้อมูล ดังรูปที่ 10

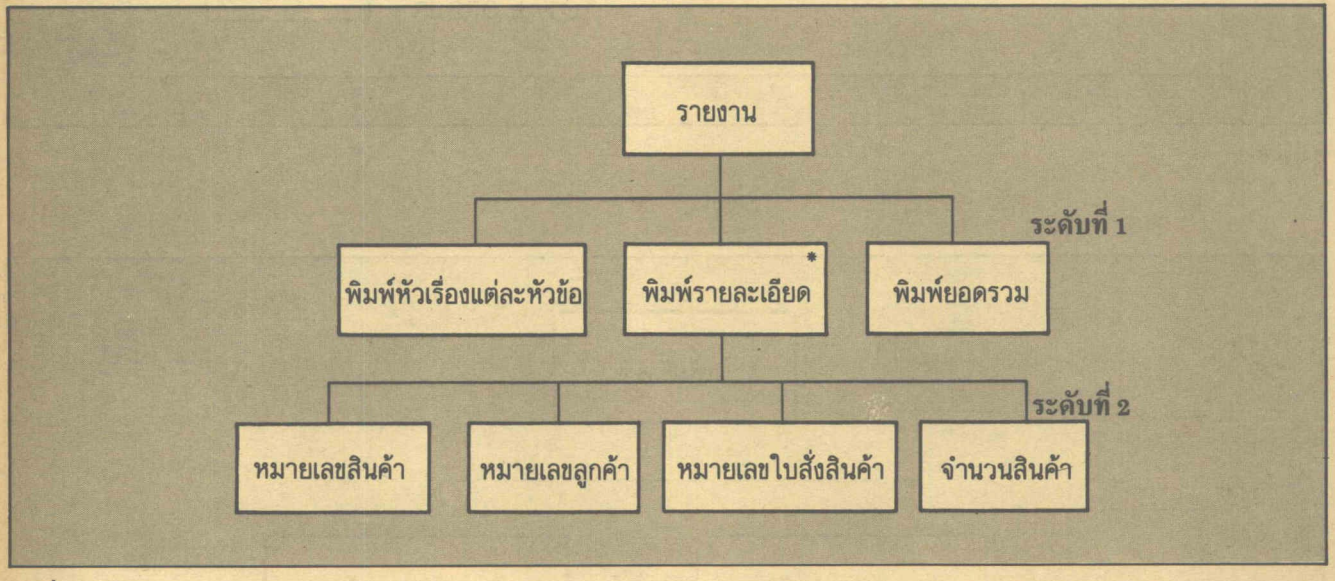
ถ่ายทอดโปรแกรมให้เป็นข้อความโครงสร้าง (Text step)

จากผังโครงสร้างโปรแกรมแบบแขนงนำมาบรรยายเป็น ข้อความโครงสร้างก่อนที่จะนำไปเขียนเป็นภาษาโปรแกรมจริงได้ดังรูปที่ 11

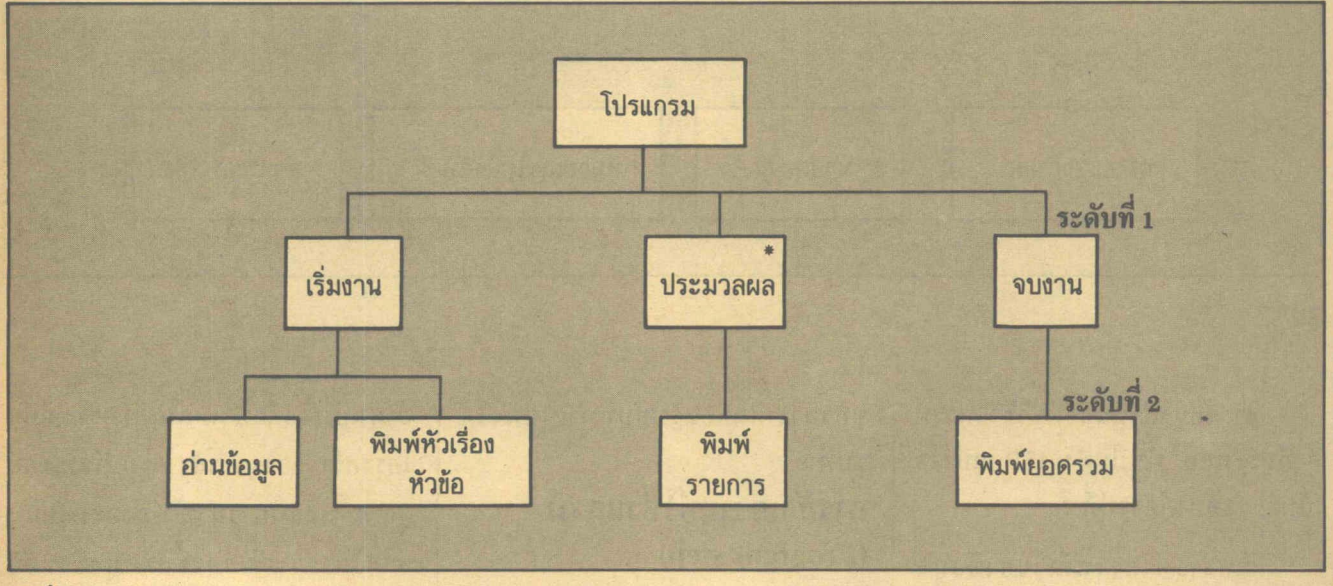
พิจารณาจากข้อความโครงสร้างเมื่อนำไปเปรียบเทียบกับภาษาโปรแกรมเครื่องที่นิยมใช้กันในปัจจุบัน จะเห็นว่าเกือบถอดเป็นภาษาได้ตรงกันแทบทุกประโยค ตัวอย่างเช่น dBASEIII ดังรูปที่ 12

หากทดสอบแนวความคิดของแจ็คสันอีกครั้งหนึ่ง ด้วยการเปลี่ยนแปลงรูปแบบรายงาน แต่ข้อมูลนำเข้ายังไม่มีการเปลี่ยนแปลง ให้พิมพ์เป็นรายกลุ่มลูกค้าเรียงตามลำดับรหัส และเรียงตามรหัสสินค้าดังรายงานรูปที่ 13

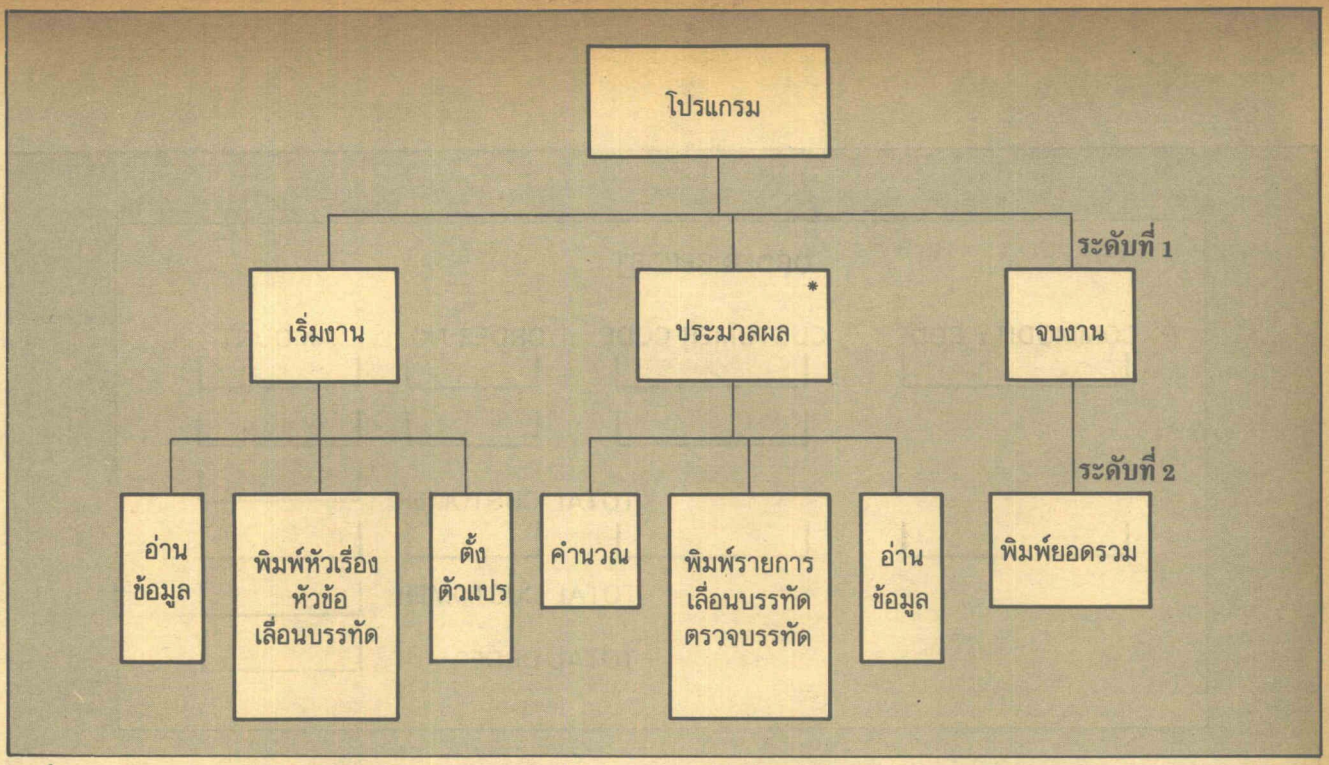
จากรายงานนำมาทำเป็นผังโดยวิเคราะห์แยกตามลำดับการเกิดและส่วนประกอบต่าง ๆ ได้ดังรูปที่ 14



รูปที่ 8



รูปที่ 9



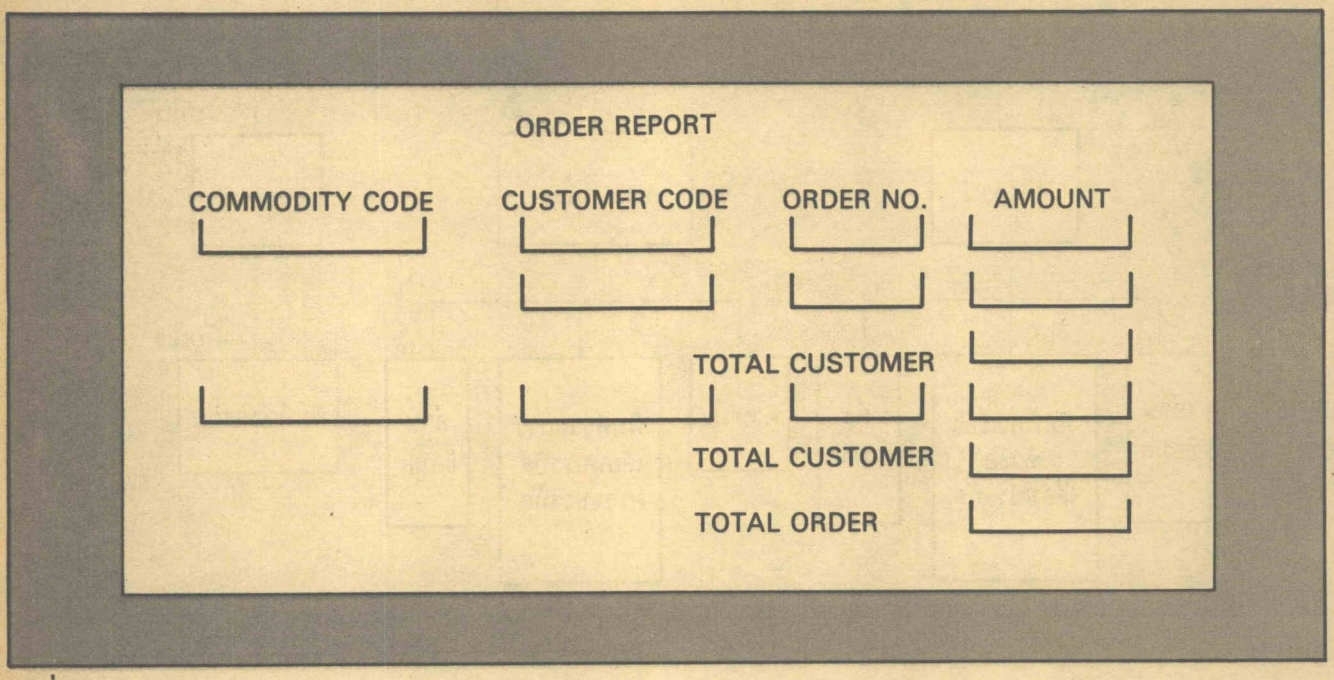
รูปที่ 10

```
เริ่มโปรแกรม seq
   อ่านข้อมูล;
   พิมพ์หัวเรื่อง;
   พิมพ์หัวข้อ;
   เลื่อนบรรทัดการพิมพ์;
   กำหนดค่าตัวแปรสำหรับการคำนวณ;
   ประมวลผลข้อมูล itr until (พบเครื่องหมายหมดข้อมูล)
      คำนวณสะสมยอดรวม;
      พิมพ์รายละเอียดของข้อมูล;
      เลื่อนบรรทัดการพิมพ์;
      ตรวจบรรทัดการพิมพ์ ขึ้นหน้าใหม่;
      อ่านข้อมูล;
   ประมวลผลข้อมูล end
   พิมพ์ยอดรวม
เริ่มโปรแกรม end
```

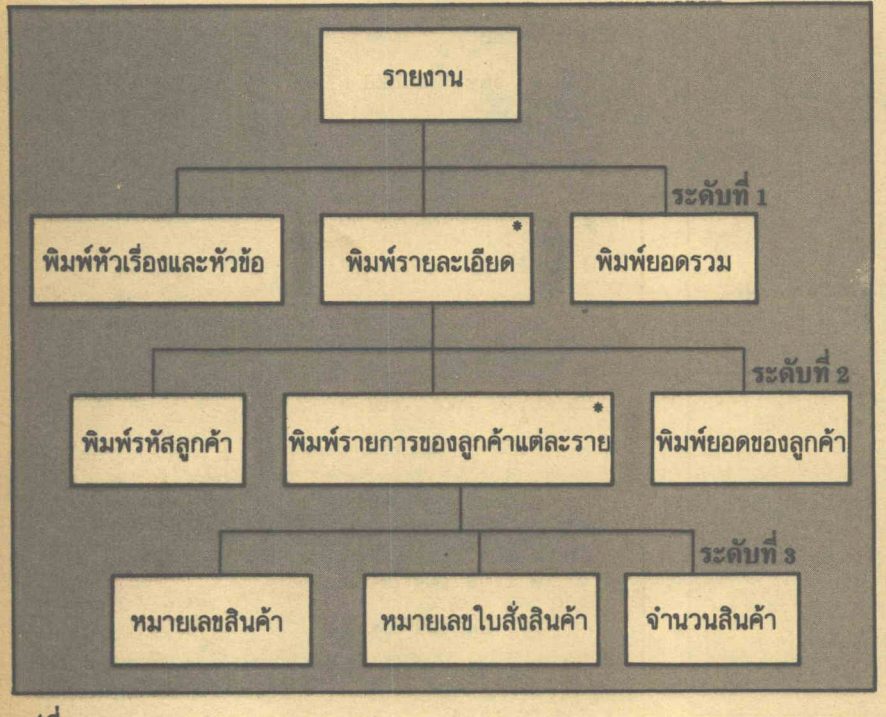
รูปที่ 11

```
*start program
set talk off
use order
@ 1,30 say " ORDER REPORT "
@ 2,10 say " COMMODITY CODE "
@ 2,30 say " CUSTOMER CODE  "
@ 2,47 say "ORDER NO."
@ 2,60 say "AMOUNT "
l=3
tamount=0
do while .not. eof()
    tamount=amount+tamount
    @ l,12 say comcode
    @ l,32 say cuscode
    @ l,48 say orderno
    @ l,62 say amount
    l=l+1
    if l>20
        @ 3,0 clear
        l=3
        wait
    endif
    skip
enddo
@ l,40 say "TOTAL"
@ l,60 say tamount
*end program
```

รูปที่ 12



รูปที่ 13



รูปที่ 14

เมื่อนำมาเปรียบเทียบกับโครงสร้างข้อมูลส่วนนำเข้าตามตัวอย่างเดิมที่ไม่ได้เปลี่ยนแปลง จะพบว่ามีข้อขัดแย้งอยู่บางประการคือ

ก. ระดับชั้นของข้อมูลไม่เท่ากัน ข้อมูลส่วนส่งออกมี 3 ระดับแต่ข้อมูลส่วนนำเข้ามี 2 ระดับ

ข. ลำดับการทำงานไม่สอดคล้องกัน ในที่นี้ข้อมูลนำเข้าอ่านทีละระเบียนไม่แยกเป็นกลุ่มของลูกค้าแต่ละรายงานบังคับให้พิมพ์เป็นกลุ่มรายการของลูกค้าแต่ละราย

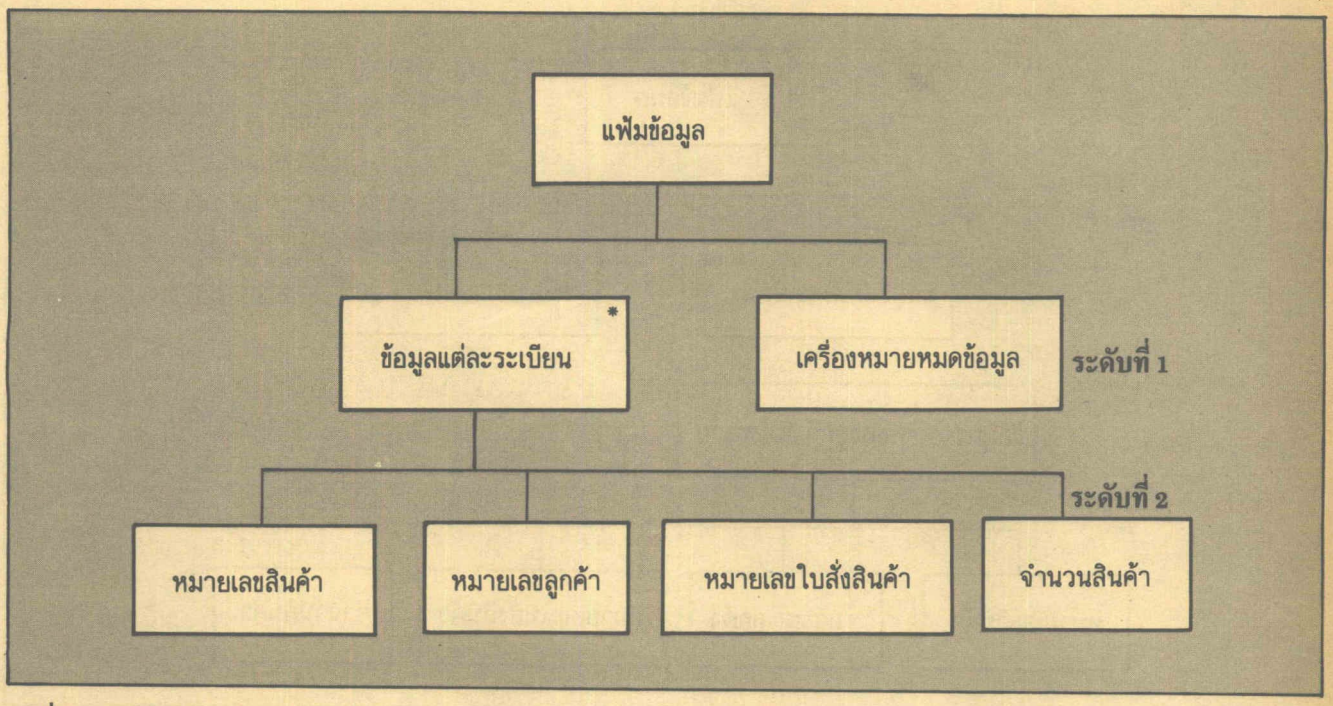
ลักษณะที่ขัดแย้งกันเช่นนี้แจ็คสันเรียกว่า **โครงสร้างที่ขัดแย้งกัน (Structure Clashes)** ดังรูปที่ 15

โครงสร้างที่ขัดแย้งกัน (Structure Clashes)

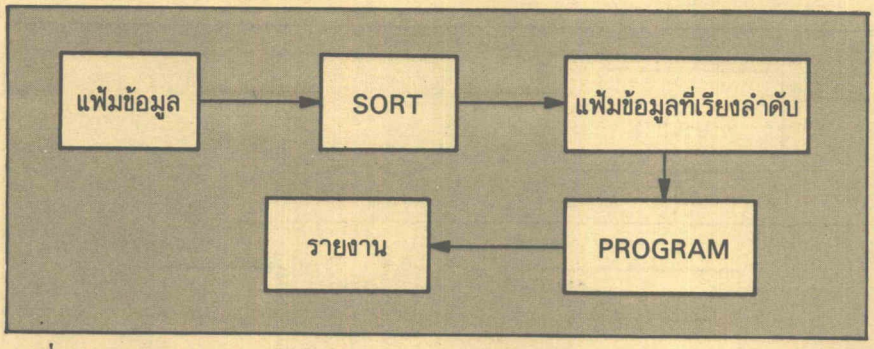
1. คำนิยาม โครงสร้างข้อมูลของส่วนนำเข้าและส่งออกกับโครงสร้างของโปรแกรมมีโครงสร้างที่ไม่สอดคล้องกันประการใดประการหนึ่งดังต่อไปนี้

ก. ได้แก่การไม่สอดคล้องกันของลำดับของการประมวลผลเทียบกันขั้นต่อขั้น หรือการส่งต่อขั้นต่อขั้นระหว่างข้อมูลนำเข้าและส่งออกไม่สอดคล้องกัน เช่นข้อมูลส่งออกต้องการรายงานเป็นกลุ่ม แต่ข้อมูลนำเข้าไม่ได้จัดทำเป็นกลุ่ม ลักษณะนี้เรียกว่า **โครงสร้างลำดับขัดแย้ง (Ordering Structure Clash)**

ข. ในลำดับชั้นเดียวกันมีโครงสร้างไม่เท่ากันหรือไม่ชัดเจนเมื่อนำมาเปรียบเทียบกัน เช่น ข้อมูลนำเข้าที่บันทึกข้อมูลลักษณะคละกัน (ตัวอย่าง รวมรหัสกับคำอธิบายรหัสไว้ให้ในเขตข้อมูลเดียวกัน) แต่ข้อมูลส่งออกต้องการแยกรหัส กับคำอธิบายรหัสไว้ต่างที่ เรียกลักษณะเช่นนี้ว่า **โครงสร้างขอบเขตขัดแย้ง (Boundary Structure Clash)**



รูปที่ 15



รูปที่ 16

2. การแก้ไข การขัดแย้งของโครงสร้างเป็นชนิดใดสามารถแก้ไขได้ด้วยวิธีการใดวิธีการหนึ่งดังต่อไปนี้

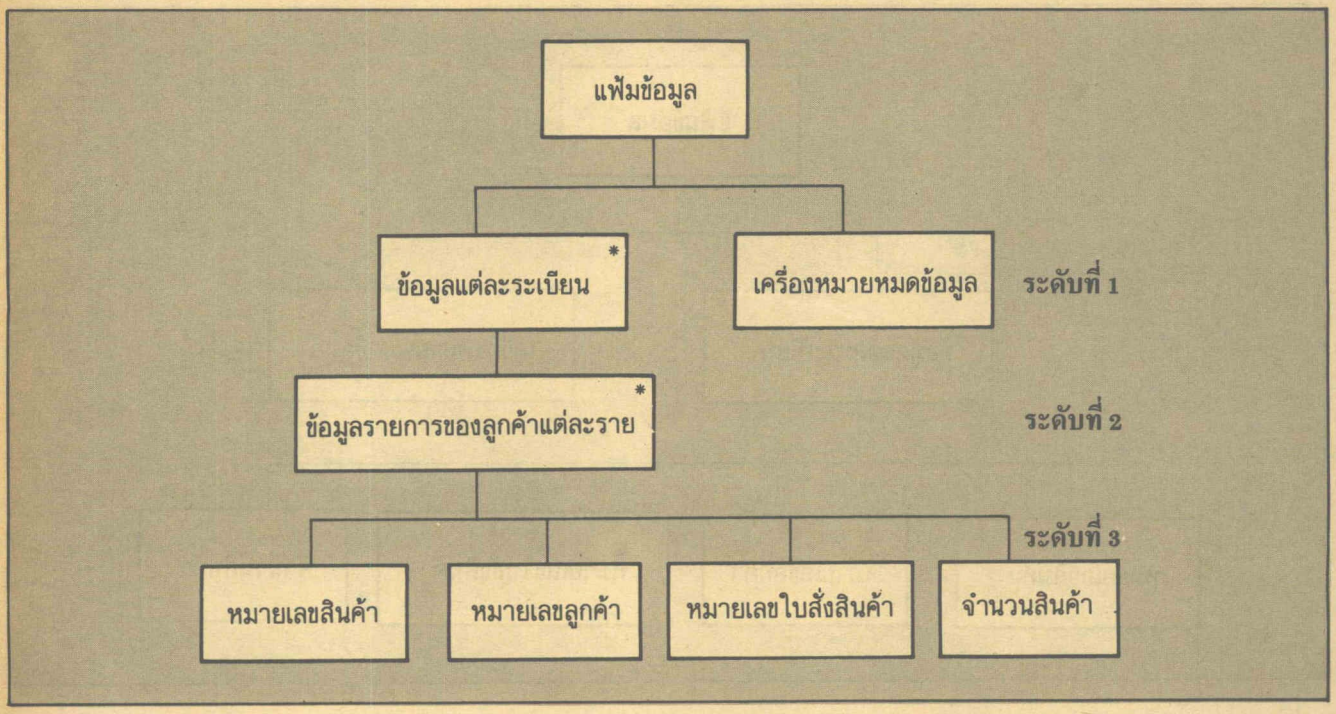
ก. จัดหรือปรับโครงสร้างข้อมูลทั้งสองคือข้อมูลส่วนนำเข้าหรือข้อมูลส่วนนำออกให้สอดคล้องกัน เช่นการทำการจัดเรียงข้อมูลนำเข้าเสียใหม่หรือแก้รูปแบบข้อมูลส่วนนำออก เช่น การแก้ไขรายงานให้สอดคล้องกับข้อมูลนำเข้า (การแก้ข้อมูลส่วนนำออกมักไม่ค่อยนิยมทำ) ดังรูปที่ 16

ข. การทำให้ข้อมูลนำเข้าเป็นแฟ้มที่อยู่ในหน่วยความจำหลัก เช่น การทำเป็นอะเรย์หรือ Table ทำให้สะดวกในการจัดรูปแบบให้สอดคล้องกับข้อมูลส่วนนำออกได้คล่องตัวมากขึ้น แต่วิธีนี้ต้องใช้เนื้อที่ในหน่วยความจำมาก

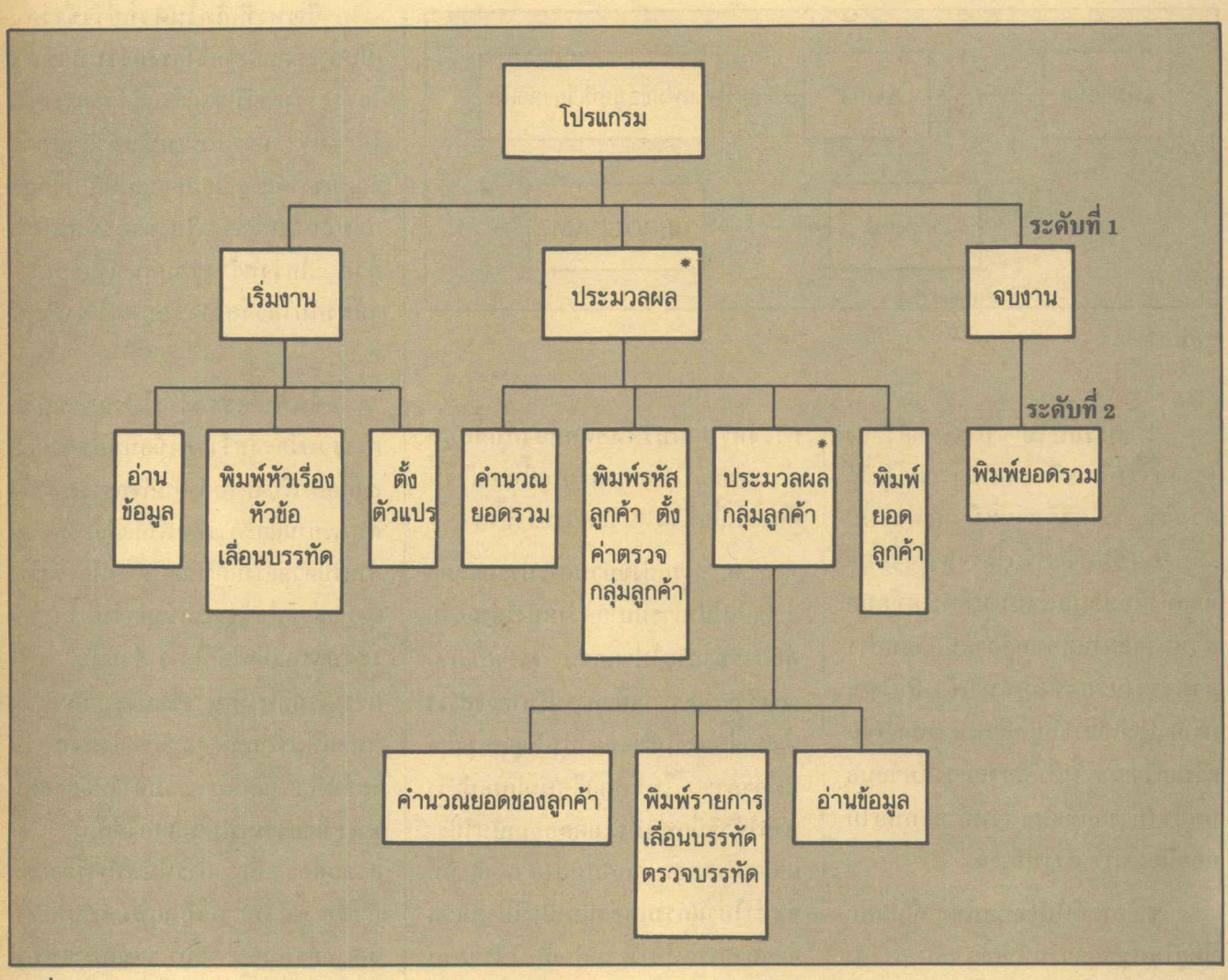
ค. การแบ่งขบวนการประมวลผลออกเป็นโปรแกรมมากกว่าหนึ่งโปรแกรม คือการทำลายโปรแกรม (Multiprogramming) ทั้งนี้เพื่อการปรับโครงสร้างที่แตกต่างกันให้สอดคล้องกันและได้ผลที่ต้องการ ในที่สุดแจ็คสันแนะนำว่าหากใช้วิธีนี้แทนที่จะแตกออกเป็นโปรแกรมแยกออกจากกันหลาย ๆ ชุด ให้จัดทำโปรแกรมเหล่านั้นเป็นโปรแกรมย่อยสามารถนำมารวมกันเป็นโปรแกรมเดียวได้

ปัญหาที่เกิดในตัวอย่างข้างต้นเป็นการขัดแย้งของโครงสร้าง ลักษณะโครงสร้างลำดับขัดแย้งแก้ด้วยการปรับโครงสร้างข้อมูลส่วนนำเข้าด้วยการจัดเรียงลำดับระเบียนข้อมูลให้เป็นข้อมูลกลุ่มลูกค้าและจัดเรียงตามรหัสสินค้าด้วย โครงสร้างข้อมูลนำเข้าจะสอดคล้องกับโครงสร้างข้อมูลนำเข้าดังรูปที่ 17

ขั้นทำโครงสร้างโปรแกรมด้วยการรวมโครงสร้างทั้งข้อมูลนำเข้าและส่งออก เรียงตามลำดับกิจกรรม คำนึงถึงการอ่านและการพิมพ์ในแต่ละขั้นตอน สำหรับตัวอย่างนี้กำหนดได้จากโครงสร้างข้อมูลส่วนนำเข้าและส่งออกว่ามี 3 ระดับ ระดับแรกแบ่งได้เป็น 3 ส่วนคือ กิจกรรมเริ่มต้น กิจกรรมประมวลผลจนหมดแฟ้มข้อมูล และกิจกรรมจบ ในระดับที่ 2 แตกกิจกรรมลงไปอีกระดับหนึ่ง **กิจกรรมเริ่มต้น**แตกได้เป็น 3 กิจกรรมย่อย คือ การพิมพ์หัวเรื่องและหัวข้อ กิจกรรมเปิดแฟ้มข้อมูลอ่านข้อมูลแรกและเลื่อนบรรทัดการพิมพ์ กิจกรรมตั้งค่าตัวแปร **กิจกรรมประมวล**


แฟ้มข้อมูล
ข้อมูลแต่ละระเบียน *
เครื่องหมายหมดข้อมูล
ระดับที่ 1
ข้อมูลรายการของลูกค้าแต่ละราย *
ระดับที่ 2
ระดับที่ 3
หมายเลขสินค้า
หมายเลขลูกค้า
หมายเลขใบสั่งสินค้า
จำนวนสินค้า

รูปที่ 17


โปรแกรม
ระดับที่ 1
เริ่มงาน
ประมวลผล *
จบงาน
ระดับที่ 2
อ่านข้อมูล
พิมพ์หัวเรื่อง หัวข้อ เลื่อนบรรทัด
ตั้งตัวแปร
คำนวณยอดรวม
พิมพ์รหัสลูกค้า ตั้งค่าตรวจกลุ่มลูกค้า
ประมวลผลกลุ่มลูกค้า *
พิมพ์ยอดลูกค้า
พิมพ์ยอดรวม
คำนวณยอดของลูกค้า
พิมพ์รายการ เลื่อนบรรทัด ตรวจบรรทัด
อ่านข้อมูล

รูปที่ 18

```
เริ่มโปรแกรม seq
    อ่านข้อมูล
    พิมพ์หัวเรื่อง;
    พิมพ์หัวข้อ;
    เลื่อนบรรทัดการพิมพ์
    กำหนดค่าตัวแปรสำหรับการคำนวณ;
    ประมวลผลข้อมูล itr until (พบเครื่องหมายหมดข้อมูล)
        คำนวณสะสมยอดรวมของทุกลูกค้า;
        ตั้งค่ากำหนดกลุ่มลูกค้า;
        พิมพ์รหัสลูกค้า;
        ประมวลผล itr whil (ขณะที่ยังไม่เปลี่ยนกลุ่มลูกค้า)
            คำนวณสะสมยอดรวมของลูกค้า;
            พิมพ์รายละเอียดของลูกค้า;
            เลื่อนบรรทัดการพิมพ์;
            ตรวจบรรทัดการพิมพ์ ขึ้นหน้าใหม่;
            อ่านข้อมูล;
        ประมวลผล end
        พิมพ์ยอดรวมลูกค้า;
        เลื่อนบรรทัดการพิมพ์;
    ประมวลผล end
    พิมพ์ยอดรวมของลูกค้าทั้งหมด;
    จบ ปิดแฟ้ม
โปรแกรม end
```

เขียนเป็นโปรแกรม dBASE III ลองตรวจสอบเปรียบเทียบกันดูจะเห็นว่าเกือบตรงกันแทบทุกประโยค

```
*start program
set talk off
use order
@ 1,30 say "ORDER REPORT"
@ 2,10 say "CUSTOMER CODE"
@ 2,30 say "COMMODITY CODE"
@ 2,47 say "ORDER NO."
@ 2,60 say "AMOUNT"
1 = 3
tamount = 0
gamount = 0
do while .not. eof( )
    gamount = amount + gamount
    ctrcode = cuscode
    @ 1,12 say cuscode
    do while ctrcode=cuscode
        tamount = amount + tamount
        @ 1,32 say comcode
        @ 1,48 say orderno
        @ 1,62 say amount
        1 = 1+1
        skip
    enddo
    @ 1,40 say "TOTAL CUSTOMER"
    @ 1,60 say tamount
    1 = 1+1
    tamount = 0
enddo
1 = 1+1
@ 1,40 say "TOTAL ORDER"
@ 1,60 say gamount
close all
*end program
```

รูปที่ 19

ผลแยกเป็นกิจกรรมย่อยระดับที่ 2 คือ การคำนวณยอดสะสมรวมทั้งหมด กิจกรรมพิมพ์รหัสลูกค้า กิจกรรมกำหนดกลุ่มลูกค้า กิจกรรมประมวลผลกลุ่มลูกค้าจนหมดกลุ่ม กิจกรรมพิมพ์ยอดของลูกค้า **กิจกรรมจบ**แยกเป็นพิมพ์ยอดรวมทั้งหมด และจบโปรแกรม โครงสร้างโปรแกรมเขียนเป็นผังได้ดังรูปที่ 18

ขั้นสุดท้ายบรรยายผังเป็นข้อความโครงสร้างซึ่งสามารถนำไปถอดเป็นภาษาคำสั่งเครื่องได้ ดังรูปที่ 19